CLAIMS:

1. A system for processing a computer instruction from a source of such instructions comprising:

a complex instruction detector having an input and an output, the input accepting computer instructions from the source and the output being indicative of whether the instruction is a member of a set of instructions,

an address generator having an input and an output, the input accepting computer instructions from the source and the output comprising an address based on the instruction,

a jump instruction generator having an input and an output, the input being in communication with the address generator output and the output comprising an instruction to jump to the address from the address generator,

an instruction selector having inputs and an output, the inputs being in communication with the jump instruction generator, the source and the complex instruction detector, the output comprising either the instruction from the source or the instruction from the jump instruction generator depending upon the output of the complex instruction detector.

2. The system of claim 1 further including a processor in communication with the output of the instruction selector.

3. The system of claim 2 further including an instruction cache in communication with the output of the instruction selector and the processor.

4. The system of claim 1 wherein the source comprises a memory.

5. The system of claim 4 wherein the memory includes machine instructions.

6. The system of claim 1 wherein the complex instruction detector, address generator, jump instruction generator and instruction selector are stored on a single computer chip.

7. The system of claim 6 wherein the address is an address in a memory located on the chip.

8. The system of claim 6 wherein the address is an address in a memory located off of the chip.

9. The system of claim 1 wherein the set of instructions comprises those instructions capable of execution by a processor in a single cycle throughput.

10. The system of claim 1 wherein the complex instruction detector includes a reserved instruction exception handler.

11. The system of claim 1 wherein the system comprises a RISC-based computer chip.

12. The system of claim 1 wherein the address generator includes a programmable logic array.

13. The system of claim 12 wherein the output of the programmable logic array depends on the opcode of the inputted instruction.

14. The system of claim 13 wherein the output of the programmable logic array depends on the operand of the inputted instruction.

15. A method of processing a computer instruction comprising:

generating an address associated with the computer instruction;

generating a jump instruction based on the address;

determining whether the computer instruction is complex; and

selecting the jump instruction or the computer instruction based on the result of the step of determining.

16. The method of claim 15 wherein the computer instruction includes an opcode and an operand.

17. The method of claim 16 wherein the step of generating an address is based on the opcode of the instruction.

18. The method of claim 15 wherein the step of determining whether the computer instruction is complex comprises determining whether the computer instruction generated a reserved instruction exception.

19. The method of claim 15 wherein the address identifies computer instructions for emulating the complex computer instruction.

20. The method of claim 15 wherein the step of generating a jump instruction comprises appending a Jump and Link instruction to the address.

21. The method of claim 15 wherein the steps of generating a jump instruction and the step of determining whether the computer instruction is complex are performed before the step of selecting.

22. The method of claim 15 wherein the address is the same for all complex instructions.

23. A method of executing a program with a processor, the processor being capable of executing a set of instructions, comprising:

providing an original instruction from a sequence of instructions comprising a program stored in a memory;

generating an address from the original instruction;

generating a jump and link instruction to the address, the jump and link instruction comprising an instruction for the processor to execute instructions at the address and then return to the instruction following the original instruction in the program;

determining whether the original instruction is a member of the set of instructions;

selecting the jump and link instruction or the original instruction based on the result of the step of determining; and

providing the selected instruction to the processor.

24. The method of claim 23 further including providing the selected instruction to an instruction cache of the processor.

25. The method of claim 24 wherein the step of providing an original instruction includes retrieving the instruction from a memory.

26. A system for processing computer instructions between a memory and a processor comprising:

a complex instruction detector connected to the memory so as to receive computer instructions from the memory and output a value indicative of whether the instruction is a member of a set of instructions,

an address generator connected to the memory so as to receive computer instructions from the memory,

a jump instruction generator connected to the address generator,

an instruction selector connected to the jump instruction generator, the memory, the complex instruction detector and the processor so as to receive: jump instructions from the jump instruction generator; computer instructions from the memory; and the value from the complex instruction detector,

whereby depending on the value from the complex instruction detector, either the jump instruction or the computer instruction is provided by the instruction selector to the processor.

27. The system of claim 26 wherein the set of instructions comprises those instructions which are capable of being executed by the processor without additional decoding.

28. The system of claim 26 wherein the set of instructions comprises those instructions which are capable of being executed by the processor in a single cycle throughput.

29. The system of claim 26 wherein the address points to other instructions which are capable of execution by the processor and which emulate the computer instruction.

30. The system of claim 26 wherein the jump instruction generator appends a jump and link instruction to the address.

31. A system for processing computer instructions comprising:

a source of complex and simple instructions, the simple instructions being capable of being executed by the processor and the complex instructions not being capable of being executed by the processor,

a complex instruction detector connected to the source and an instruction selector, the complex instruction detector receiving computer instructions from the source and providing a value indicative of whether a received instruction is complex or simple,

an address generator connected to the source and a jump instruction generator, the address generator receiving computer instructions from the source and, if a received instruction is complex, providing an address in a memory containing emulation instructions, whereby the emulation instructions are simple instructions and emulate the intended function of the complex instruction,

the jump instruction generator being connected to the address generator and the instruction selector, the jump instruction generator receiving addresses from the address generator and providing a jump and link instructions to the addresses,

the instruction selector being connected to the jump instruction generator and the complex instruction detector such that the instruction selector provides a jump and link instruction from the jump instruction generator if the instruction received from the source is complex, or provides the instruction if the instruction received from the source is simple, and

a processor for receiving the instructions from the instruction selector.

32. The system of claim 31 further comprising an instruction cache disposed between the instruction selector and the processor.

33. The system of claim 31 wherein the address generator provides a first address in response to a first complex instruction and a second address in response to a second instruction, the first and second addresses being different.

34. The system of claim 31 wherein the complex instruction detector uses routines associated with reserved instruction exceptions.